

REMARKS

Status

Claims 1-11, 13, 14, 30, and 49-785 are pending. There are a total of 51 pending claims, five of which are independent.

This Applicants acknowledge with appreciation the withdrawal of previous objection to the Specification under 37 C.F.R. § 1.75(d)(1) and the withdrawal of the rejections of claims 1-11, 13, 68, and 70 under 35 U.S.C. § 103(a). However, the Office Action of September 3, 2009 (hereinafter, “the Office Action”) again rejects the claims, again citing new prior art and presenting new grounds of rejection.

The Office Action does not explicitly specify grounds for rejecting claims 49-66, stating only that, claims 49-50 correspond to claims 13-14, claims 51-63 correspond to claims 1-11 and 13-14, and claims 64-66 correspond to claims 30, 13, and 14. Applicants disagree and note that claims 49-50 do not relate to claims 13-14, however the other claims do relate to a significant degree as indicated. To the extent that they do relate, it should be understood that the claims are not intended to necessarily have identical scope, i.e., certain differences in scope are intentional and it should not be assumed that a finding of patentability or unpatentability of one claim necessarily results in the same finding for a corresponding claim.

Claim Rejections – Mahalingam, Vega, and Vepa

Claims 1-10, 68, and 70 stand rejected under 35 U.S.C. § 103(a) for being unpatentable over U.S. Patent 6,208,616 to **Mahalingam et al.** (hereinafter referred to as “Mahalingam”) in view of U.S. Patent 7,136,800 issued to **Vega** (hereinafter referred to as “Vega”) and U.S. Patent 6,567,377 issued to **Vepa et al.** (hereinafter referred to as “Vepa”) (Office Action, page 4). The Office Action further states that claims 51-63 correspond to claims 1-11 and 13, 14 and that they are rejected for the same reasons (Office Action, page 13). Applicants interpret this to mean that claims 51-60 are also rejected for being unpatentable under 35 U.S.C. § 103(a) over Mahalingam in view of Vega and Vepa. Applicants respectfully traverse for the reasons explained below.

1. None of the references teach or suggest “based on . . . VM-specific information, selecting a NIC from the plurality of NICs and transferring the outgoing data frame to the computer network over the selected NIC”

Mahalingam:

Mahalingam teaches a system for sending outgoing network frames to a second NIC if the first NIC fails. Mahalingam also includes a mechanism for testing the functionality of the NICs that would trigger the fail-over. It appears to Applicants that Mahalingam was cited in error however, since the Office Action cites Mahalingam for showing “obtaining access by a NIC manager to outgoing data frame” in “Fig. 9-10,” and for showing “based on NIC management information . . . selecting a NIC from the plurality of NICs” in “abstract, paragraph 98, and Fig. 10” (Office Action, page 4) yet Mahalingam does not have numbered paragraphs, and does not have a Figure 10. However, Applicants agree that Mahalingam does show load sharing and fault tolerance using a plurality of NICs to communicate with a plurality of routers or switches on a network. Applicants also agree that Mahalingam does not show “receiving or using VM-specific information in the decision making process” (Office Action, page 4, lines 21-22).¹

Vega:

Vega teaches distributing processor resources among a plurality of virtual machines according to a particular algorithm. The Office Action cites Vega for showing “using VM-specific information (col. 3 ln. 65-col. 4, ln. 7) to manage the host computer’s resources” (Office Action, sentence spanning pages 4-5). The Office Action further states that “[a]lthough Vega is directed to allocating processor time, one skilled in the art understands that a NIC is a simply [*sic*] a computer resource and that similar allocation methods can be used” (Office Action, page 5, lines 1-3).

Applicants agree that a person skilled in the art would understand that a NIC is a computer resource. However, one would also have understood that the mechanism and algorithm described by Vega for allocating and distributing processor time could not be used to allocate or distribute a NIC resource. As explained in more detail in the paragraph bridging pages 14 and 15 in the Amendment submitted on July 3, 2008, and as one example of the differences between processor scheduling and NIC scheduling, when a VM is descheduled so that a different VM can execute, it is like hitting a “pause” button, in that there is no activity in that VM during the time it is descheduled. However, when a VM is denied access to a NIC, the

¹ Line numberings for referenced documents count every printed line only, including headings, but not including the page header, unless the referenced document is already includes line numberings, as with issued U.S. Patents.

VM may still be executing and in fact be generating outgoing data frames, which queue up over time due to the non-availability of the assigned NIC.

Furthermore, the processor allocation is performed by the kernel process and as such has access to VM-specific information such as relative priorities. As explained, for example in paragraph 77 on page 25 of the Application as filed, using a loadable module or driver to implement NIC teaming functionality, as was known prior to the present invention, places barriers to access of VM-specific information that is useful in making NIC selection decisions. That is, the VM-specific information is not accessible by prior art loadable modules or drivers implementing NIC teaming functionality

Vepa:

Vepa discloses a “dynamic access software element [that] consists of a protocol and media access control (MAC) driver” (col. 9, lines 11-13) that implements a NIC load balancing scheme (col. 9, lines 17-18). The MAC driver is “a software element that is inserted between the protocol stack (e.g., the network layer) in the server computer system and the NIC drivers” (col. 12, lines 11-14). The Office Action cites Vepa for teaching “a load balancing scheme for selecting physical NICs that takes into consideration the source of the data (col. 3 ln. 40 – col. 4 ln. 17)” (Office Action page 5, lines 4-5). The Office Action appears to suggest that the source described by Vepa “represents the virtual machine” (Office Action, page 5, lines 5-6), but Vepa does not mention virtual machines. Furthermore, in the portions of Vepa referenced by the Office Action, Vepa teaches several load balancing algorithms that take into consideration only information from the packets, i.e., without any data analogous to the VM-specific data set forth in the claims.

Specifically, Vepa teaches that the selected NIC can be based on the following data:

<i>Data used by Vepa to select NIC</i>	<i>Citation</i>
IP Address of the outgoing data packet	col. 3, lines 56-57
Number of NICs	col. 3, lines 56, 60-61
Destination TCP port of the outgoing data packet	col. 3, lines 66-67, col. 4, lines 8-10
Source TCP port of the outgoing data packet	col. 3, lines 1, 3, 7-8

Thus all information being used to select a NIC includes either NIC-specific information (i.e., the number of NICs) or information that characterizes the data frame, i.e., its destination IP address or its source or destination ports. Contrary to assertions in the Office Action (quoted

above), Applications could not find any suggestion in Vepa that any VM-specific information, or any information analogous thereto, is used to select a NIC to send an outgoing data frame.

To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). Since none of the cited references teach or suggest using VM-specific information when selecting a NIC over which an outgoing data frame is to be sent, Applicants respectfully submit that the Office Action fails to make out a case of prima facie obviousness under 35 U.S.C. § 103(a), and therefore claims 1-10, 51-63, 68, and 70 should be allowed. Since claims 11, 13-15, and 67-70 each depend on either claim 1 or claim 51, Applicants respectfully submit that these claims should likewise be allowed for at least the reasons discussed above with respect to claims 1 and 51.

2. The prior art lacks a suggestion to combine and/or modify the claims as proposed in the Office Action.

The Office Action states that, “[t]he combination is merely the combination of known elements according to their established function in order to yield a predictable result” (Office Action, page 5, lines 9-10). Applicants respectfully disagree with this assertion, and further note that the Office Action has not identified specific motivation that was present in the prior art that would have led a person of ordinary skill in the art to combine and/or modify the references as proposed.

Initially, Applicants respectfully note that in *KSR*, infra, obviousness was concluded where “all the claimed elements were known in the prior art and one skilled in the art could have combined the elements as claimed by known methods with no change in their respective functions, and the combination would have yielded nothing more than predictable results to one of ordinary skill in the art” (MPEP 2143.02, citing, *inter alia*, *KSR International Co. v. Teleflex Inc.*, 82 USPQ2d 1385, 1395 (2007) (emphasis added)). In the present case, the elements cannot be combined “with no change in their respective functions” since none of the references teach selecting a NIC for an outgoing data frame using VM-specific information or anything analogous thereto.

Thus, under *KSR*, it is not enough that the combination yield a predictable result to establish *prima facie* obviousness, but in addition, the functions of each component being brought together must not change when making the combination. Summarizing the rejection, Mahalingam was cited for showing load balancing and fail-over for a plurality of NICs attached to a server, Vega was cited for showing processor resource allocation to a plurality of VMs, and Vepa was cited for showing algorithms for balancing NIC resources. The proposed combination applies Vepa's or Vega's resource distribution algorithm to the NIC load balancing and failover functions of Mahalingam or Vepa. However, Vepa relies on VLAN identifiers from applications and virtual MAC addresses to implement its load balancing scheme (col. 9, lines 11-27), each virtual MAC corresponding to a different VLAN.

Mahalingam and Vepa:

Mahalingam and Vepa each describe NIC teaming to implement redundancy and load balancing. As described in each application, the software component is inserted between a network stack and NIC drivers. Mahalingam shows this in Figure 3 where the component in question is referred to as a "MULTISPAN" process (shown at 110 in Figure 3 and described at col. 6, lines 25-49. Mahalingam discusses the method for loading the MULTISPAN component, which involves making a patch to the server code as described in col. 8, lines 8-15. It is clear from the ensuing description that Mahalingam is a separate process and does not have access to "source" information analogous to VM-specific information described in the present Application.

Likewise, Vepa discloses, as explained above, a "dynamic access software element [that] consists of a protocol and media access control (MAC) driver" (col. 9, lines 11-13) that implements a NIC load balancing scheme (col. 9, lines 17-18). The MAC driver is "a software element that is inserted between the protocol stack (e.g., the network layer) in the server computer system and the NIC drivers" (col. 12, lines 11-14). Again, it is clear that Vepa is a driver component that does not have access to "source" information analogous to VM-specific information described in the present Application.

That Mahalingam and Vepa would require significant modification to implement a NIC redundancy and load sharing solution is also made clear in the present Application, e.g., in paragraphs 82 or 136, which describes the integration of the NIC manager with the kernel to enable the NIC selection algorithm to take into consideration VM-specific information. Thus,

using VM-specific information or analogous information is not a matter of simply adding one more variable into the NIC selection algorithm. Accessing VM-specific information, or information analogous thereto, that is known to the kernel and not to the loadable driver or the separate process that controls the NIC teaming functionality as implemented by Mahalingam and Vepa is not possible.

Vega:

Vega's processor time to VM allocation algorithm cannot be applied to NICs because (a) Vega does not teach or suggest selecting a resource from a plurality of like resources based on VM-specific information (Vega deals *exclusively* with allocating a percentage (or fraction) of a unitary resource, i.e., processor capacity), and (b) fundamental differences between processor as a resource and NICs, as a resource, would not lead a person having ordinary skill in the art to look to processor scheduling techniques when determining how to select a NIC from a plurality of NICs when sending an outgoing network data frame. Thus, the algorithm of Vega would require modification in order for it to be applied to NIC selection based on VM-specific information.

Fundamentally, a network data frame received from a VM is different from a thread waiting to be allocated processor time for a number of reasons. First, when you deschedule a thread from a processor, it stops functioning, and therefore stops sending out data access requests etc. However, when you "deschedule" a VM from accessing the network, the VM continues to operate and can continue to generate outgoing dataframes. Therefore, the queue of outgoing dataframes continues to increase, and can do so indefinitely. If the particular VM is at a lower priority and therefore gets less bandwidth, the outgoing queue can grow faster than data frames can be sent out, causing increased and unacceptable network latencies. If latencies grow too much, the sending VM will assume the recipient never received them, even though they are queued. This will cause the VM to re-send the queued data frames, further exacerbating the situation. These are issues that can never arise in the context of scheduling processes or threads on a processor, i.e., the art of Vega. Typically, any computer system only has a limited number of threads running concurrently (either simultaneously on separate processor cores or time-wise interleaved). Managing a limited number of threads in a processor queue is therefore distinctly different. The algorithm described in Vega for assigning processor time to VMs is therefore simply not usable for performing NIC selection based on VM-specific information.

By the way, this shows why a VM is a different type of client than an application running directly on server hardware. Applications can be prevented from sending out network data when the network queue is full. However, applications running in a virtual machine are sending data to a virtual NIC that is connected to a virtual network switch within the host computer system. Referring to Figure 3 of the present Application, the Application 260 residing in VM 200 writes to virtual NIC (VNIC) 280, which may not have any frames backed up in its queue. The VNIC 280 then transfers the data to the Kernel 600, which manages a virtual network switch for routing network frames between VMs on the same host or to different hosts over a physical network. The kernel must handle the data received from VM 200. Kernel 600 cannot simply indicate to guest application 260 that the “physical NICs 180A-180C are busy, please wait before sending more data.” Other strategies are therefore required to handle this influx while at the same time guaranteeing some Quality of Service to the VMs. This is a problem that, prior to this Application, has not been identified or addressed.

The Office Action cites, as motivation to combine the references, “one of ordinary skill in the art would understand that it may be advantageous to use NIC and sender information (VM specific information) to select a physical network interface for the purpose of load balancing.” However, as stated above, both Mahalingam and Vepa disclose load balancing, therefore there would not have been any need to use NIC and sender (VM-specific) information. The only motivation to use VM-specific (or analogous) information in a NIC selection algorithm would be to apply Quality of Service (QoS) to the VMs’ network access. Such motivation can only have come from Applicants’ own disclosure. Since the Office Action does not provide a clear articulation as to why the cited references would have been combined, at the time the invention was made, in such a manner as to arrive at the claimed invention, it can be inferred that the Examiner is instead relying on improper hindsight analysis.² Accordingly, Applicants respectfully submit that claims 1 and 51 are allowable over the prior art of record, and early allowance of the same is respectfully requested. Claims 2-10, 51-60, 68, and 70 depend from either claim 1 or claim 51, and should therefore be allowed for at least the same reasons as claims 1 and 51.

² *In re Kahn*, 441 F. 3d 977, 998 (Fed. Cir. 2006) (“When the Board does not explain the motivation, or the suggestion or teaching, that would have led the skilled artisan at the time of the invention to the claimed combination as a whole, we infer that the Board used hindsight to conclude that the invention was obvious” (citation omitted).)

Claim Rejections – Macchiano, Vepa, and Mahalingam

Claims 30 and 64 stand rejected under 35 U.S.C. Â§ 103(a) for being unpatentable over U.S. Patent 7,111,303 issued to Macchiano et al. (hereinafter, “Macchiano”), in view of Vega and Mahalingam. Claim 64 is a “Beauregard” style claim that generally corresponds in scope to the claim 30, which is a method claim.

Macchiano is directed to a virtual networking system for connecting a plurality of virtual machines (referred to by Macchiano as “user portions” (see col. 4, lines 50-52)) running on a single physical computer system. Each virtual machine has a corresponding virtual NIC 42, 44 as shown in Figure 1.

The Office Action misconstrues the Macchiano reference. Specifically, the Office Action states that,

“Macchiano also discloses, ‘the virtual computer system also comprising a first physical network interface card (NIC) and a second physical NIC for connecting to the computer network’ as describing each user portion having a virtual NIC and the computer system may also contain multiple physical NICs (see col. 3 lines 56-58 and Fig. 1 comp. 42-44; col. 5 lines 4-6)” (Office Action, page 11).

Applicants have carefully reviewed the indicated portions of Macchiano and find no support there for a plurality of physical NICs. Macchiano does describe (referring to Macchiano, Fig. 1) a computer system 10 having two guest VMs 12, 14, each having a corresponding virtual NIC 42, 44. The indicated portions of the specification of Macchiano likewise show only virtual NICs, not physical NICs.

The Office Action additionally states that Macchiano discloses, in col. 3, lines 60-66, “determining which VM within the virtual computer system is involved in the requested data transfer; and if the first VM is involved . . . transferring the data over the first NIC” (Office Action, page 11, lines 17-20). Applicants could not find such a disclosure in Macchiano at the identified location or any other place within Macchiano.

Furthermore, Macchiano does not disclose “transferring the outgoing data frame over an available one of the physical NICs if the one of the first VMs having higher priority provided the outgoing data frame; or discarding the outgoing data frame if the other of the VMs provided the outgoing data frame” as now set forth in claims 30 and 64. Furthermore, none of the remaining cited references teach this feature.

For the reasons mentioned above, Mahalingam and Vega fail to overcome the deficiencies of Macchiano. For at least the above-stated reasons, Applicants respectfully submit that claims 30 and 64, and claims depending therefrom, are allowable and therefore request reconsideration in view of the amendments now made to these claims.

Remaining Claims

Dependent claims 11, 13, 14, 49, 61-63, 65-67, and 69 stand rejected under various combinations of Mahalingam, Vega, Vepa, Macchiano, U.S. Patent 6,810,421 issued to Ishizaki et al., and U.S. Patent 7,203,944 issued to Rietschote et al. However, none of the cited references overcome the deficiencies noted above with reference to the references asserted against independent claims 1, 30, 51, and 64.

Ishizaki:

Ishizaki is cited for showing “that discarding packets can reduce administration overhead (col. 2 ln. 5-6)” (Office Action, page 9, lines 11-14). Furthermore, the Office Action states:

Applicant argues that Ishizaki discards packets received and therefore does not teach discarding outgoing data frames. This argument is not persuasive. Ishizaki clearly discloses discarding a packet if there is no match and transferring it if there is a match (col. 8, ln. 42-53). It is a reasonable interpretation that a packet about to be transferred it is an “outgoing data packet”. Ishizaki discloses deciding whether or not to transfer a packet (i.e. outgoing data packet) by looking at a routing table and discarding the packet if a decision is made not to transfer. Thus Ishizaki teaches “discarding the outgoing data frame if a decision is made not to transfer the outgoing data frame” as recited by the claims” (Office Action, pages 2-3).

Applicants respectfully disagree. It is not a reasonable interpretation that “transferred” means “outgoing,” particularly when Ishizaki otherwise makes clear that the packet being discarded is incoming. Hence, Ishizaki merely describes discarding data packets that are received (i.e., is an *incoming* packet) by a network node but are not addressed to the recipient that is managed by the node. This is not the same as discarding an outgoing data frame as set forth in claims 67 and 69.

Since Ishizaki does not cure the deficiencies of Mahalingam, Vega, and Carollo, and none of the cited references teach or suggest each and every limitation set forth in the claims, Applicants respectfully submit that claims 67 and 69 should be allowed. Reconsideration and

withdrawal of the outstanding rejections with respect to claims 67 and 69 is therefore respectfully requested.

Rietschote:

Rietschote is cited for “suspending a virtual machine to balance load” (Office Action, page 10, lines 6-7). Rietschote is directed to a method for migrating a virtual machine for the purpose of load balancing (see, Rietschote, e.g., the title). Rietschote does not suggest selecting a NIC from a plurality of NICs based on VM-specific information for transferring an outgoing data frame.

New claims 71-85 are allowable for at least the same reasons mentioned above with reference to claim 1.

Conclusion

Applicants respectfully request reconsideration of the outstanding rejections in light of the above arguments and a Notice of Allowance is respectfully and earnestly requested. The Examiner is invited to contact the undersigned at 650-427-2390 to discuss any additional changes the Examiner may feel is necessary in light of this Amendment.

Date: February 3, 2010

Respectfully submitted,

/Leonard Heyman/
Leonard Heyman
Reg. No. 40,418
Customer Number 36378